

## Objective C Programming

### Lab 1

1. Write a “City” class, which inherits from NSObject. Your class should contain the following:

Variables:

name, age, population.

Instance methods:

setName:age:population (single method) which set city’s name, age and population.  
getName, getAge, getPopulation which return city’s name, age and population, respectfully.

nextDay which adds a random number to city’s population, then subtracts a random number from city’s population. Figure out a way to generate random numbers yourself.

2. Create an instance of City class, set its name, age and population as you want.

3. Write a for-loop (if in doubt how to do it – google or use Xcode’s help system) for 10 steps. Each step send ‘nextDay’ message to your object and print out the population.

4. Write a “Metropolis” class. It should contain the following:

Variable:

array of 10 cities.

Instance method:

createCity:atIndex:withPopulation: (single method) which creates a city with first parameter being a name at index (from the second parameter) and sets its population to that of third parameter. So, you should be able to do this:

```
[myMetropolis createCity: @"Almaty" atIndex: 2 withPopulation: 1500000]
```

5. Create an instance of Metropolis class and create all 10 cities.

6. Write a for-loop for 10 steps. Each step send ‘nextDay’ message to different city in your metropolis. So, 1<sup>st</sup> step – ‘nextDay’ to 1<sup>st</sup> city, 2<sup>nd</sup> step – ‘nextDay’ to 2<sup>nd</sup> city, etc. Print out the population of affected city each step.

7. Create a class called “BetterMetropolis”. It should inherit from “Metropolis” class and include the method you wrote in step 6 as an instance method.

8. Write a “Calculator” class, which will be a simple four-function calculator you can use to add, multiply, subtract, and divide numbers. Similar to a regular calculator, this one must keep track of the running total, or what’s usually called the accumulator. So methods must let you set the accumulator to a specific value, clear it (or set it to zero), and retrieve its value when you’re done. Here is a class declaration for you:

```
@interface Calculator: NSObject

-(void) setAccumulator: (double) value;
-(void) clear;
-(double) accumulator;

-(void) add: (double) value;
-(void) subtract: (double) value;
-(void) multiply: (double) value;
-(void) divide: (double) value;

@end
```

The method `accumulator` is used to return accumulator value.

9. Create an instance of “Calculator”. Embed it into “BetterMetropolis” and use it to keep track of all population changes. So, every time any city in your metropolis get a population change, the calculator of the metropolis adds or subtracts the same number from its accumulator.